AOL DATA STRUCTURE KELOMPOK 5 LC95

Darren Nathanael Bhekti - 2702302574
Jose Antonio Marco Haryanto - 2702299680
Muhammad Bachtiar Rasyid - 2702244750
Yongky Alexander Tristan - 2702254676

Lecturer : Dimas Elang Setyoko, S.Kom., M.Cs.

Question :

1) **(10 points) Background consists of team's planning and the reason of choosing the algorithm related with the suitability between case and algorithm.**

2) **(10 points) Literature Review consists of theory of related/chosen data structure and general theory (theory that align with the case of the program).**

3) **(10 points) Benefits consists of benefits of choosing the al.**

4) **(10 points) Result consists of all feature screenshot, the program code (is presented in written form, not screenshot form) and give the coding documentation of the program (the code/function with human language).**

Number 1 :

Our efforts to develop an effective program for managing high school graduation databases required an in-depth analysis of the various data structures and algorithms available. The program we designed has several main features, including entering new student data into the database (insert), displaying student data using various traversal methods such as pre-order, in-order, and post-order (view), viewing student data based on certain subjects (view certain), searching for student data based on specific criteria such as NISN (view specific), deleting student data based on NISN (delete), as well as a feature to exit the application (exit).

The choice of AVL Tree as the main algorithm in managing this program's data was not done without strong reasons. AVL Tree is a data structure that offers various advantages that perfectly suit the needs and characteristics of the applications we develop. The reason we use avl tree

AVL Tree automatically maintains its balance after each insert and delete operation. This balance is very important because it ensures that every operation on the tree, be it search, insertion, or deletion, has a time complexity of O(log n). In the context of a graduation database whose data can continue to increase over time, stability and consistency of performance is very much needed. With AVL Tree, we can avoid the problems that often

occur with unbalanced search binary trees, such as search times increasing exponentially due to an imbalance in the tree.

AVL Tree's ability to search data quickly and efficiently is another strong reason. The main feature of our program is the ability to search for student data based on certain criteria, such as NISN or subjects. AVL Tree, with its balancing properties, allows data searches to be performed very efficiently. This is especially important for view certain and view specific features, which require data searches to be carried out quickly to provide a good user experience.

AVL Tree also supports various traversal methods (pre-order, in-order, post-order) used in the data view feature. With AVL Tree, traversal can be performed quickly and efficiently, allowing users to view data in the desired order without performance bottlenecks. In-order traversal, for example, is very useful for displaying student data in alphabetical order or based on NISN, which is very helpful in analyzing and presenting more structured data. And several other reasons that we cannot mention one by one

Thus, the use of AVL Tree allows us to efficiently fulfill all functional requirements of the program and provide an optimal user experience. This algorithm is proven to be able to overcome various existing technical challenges, making it the best choice for managing high school student graduation databases. AVL Tree provides the perfect combination of performance, efficiency, and the ability to handle dynamic data, making it a strong foundation for the applications we develop.

Number 2 :

What is AVL Tree?

AVL tree is a tree for modified binary search with left and right subtrees that have the same height or at least the difference between the two subtrees is equal to 1 or with a balance factor of 0 or not less than -1 or not more than 1. Advantages of trees AVL includes optimization of data retrieval, especially for trees that are skewed to the left or right, so that searching becomes easier if the tree is balanced. A tree can be left or right skewed, when elements are added and removed sequentially and the order of the elements is unknown. Most current applications carry out operations of adding and deleting elements continuously without any clear order, therefore in the AVL tree there is a feature that balances the tree directly which can make data searches more effective.

Important points about AVL Tree:

1. Balanced Factor: The Balanced Factor of each node in the AVL Tree is equal to the difference between the height of the left subtree and the height of the right subtree. The Balanced Factor can be -1, 0, or 1 for each node.

2. Rotation: If an insertion or deletion operation causes an imbalance in the tree, rotation is performed to restore balance. There are four types of rotation: left rotation, right rotation, left/right rotation (double rotation), and left/right rotation (double rotation).

3. Height Balanced: AVL Tree has a balanced height, where the left and right subtrees of each node differ in height by at most one. This characteristic keeps the tree balanced and maintains performance.

4. Operations: Basic operations on an AVL Tree include insertion, deletion, and search. This operation is similar to a regular binary search tree, but includes an additional step to balance the tree after each operation to preserve AVL properties.

5. Advantages: AVL Tree provides efficient search, insert, and delete operations with a worst-case time complexity of O(log n). They are suitable for applications where these operations are performed frequently and require a balanced tree structure.

Operations on AVL-Tree:

1. Element search operation: This search operation is more or less the same as that in a binary search tree because AVL is a modified BST by adding a function to regulate the balance of the tree.

2. Element addition operation: this element addition operation makes this tree have the possibility of becoming imbalanced due to a balance factor that does not meet the criteria, namely -1, 0 or 1.

3. Element insertion operation: Element deletion in the Binary Search Tree is the most complex of these three operations. When deleting this element, there are 3 cases that we must pay attention to, including:

    a. Deleting leaves: deletes nodes that do not have subtrees.
    b. Deleting a node that has a left/right uptree: delete it and replace it with its uptree
    c. Deleting a node that has 2 subtrees: overwrite the node you want to delete with the node that has the largest key in the left subtree.

reference :

- https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2006-2007/Makalah/Makalah0607-24.pdf
- https://www.geeksforgeeks.org/introduction-to-avl-tree/
- https://www.w3schools.com/dsa/dsa_data_avltrees.php
- https://student-activity.binus.ac.id/himmat/2022/10/apa-itu-avl-tree/
- https://www.javatpoint.com/avl-tree

Number 3 :

Benefit Using AVL Tree :

1. Auto Balance Settings

   AVL Tree is a binary search tree that automatically maintains balance after each addition (insert) and deletion (delete) operation. This balance ensures that the tree depth remains minimal, which in turn optimizes the time required for search operations and data manipulation. In the context of our application, this means that searching and retrieving student data based on NISN or other criteria can be done efficiently with a time complexity of O(log n).

2. Data Search Efficiency

   The balanced nature of AVL Tree supports efficient data search. Tree balance minimizes the number of steps required to find certain data, such as searching by student identification number (NISN). This is important in graduation database management applications, where users often need to quickly access specific information about students.

3. Support for Multiple Type Traversal

   AVL Tree supports various traversal methods such as pre-order, in-order, and post-order. These methods allow users to display student data in different orders, according to data analysis and presentation needs. For example, in-order traversal can be used to display student data in alphabetical order by name, or by NISN.

4. Optimal Performance for Dynamic Data

   In scenarios where the database continues to grow over time, AVL Tree can still provide consistent performance. Its ability to maintain balance ensures that insert and delete operations remain efficient, even when the amount of data increases dynamically. This is especially relevant for our application, where the student graduation database can continue to grow every year.

5. Support for Insert and Delete Operations

   AVL Tree not only excels in search, but also in insert and delete operations. A balanced tree structure ensures that after each addition or deletion of data, the tree will be reset automatically to maintain its balance. This is important in the context of our application which requires the ability to add new student data and delete data that is no longer relevant efficiently.

6. Proven Algorithm Choices

   AVL Tree has been a proven choice in computer science for efficient data management. Its properties and guaranteed performance make it a reliable choice for applications that require optimal data management. Choosing AVL Tree for your graduate student database management program not only optimizes current application performance, but also provides a strong foundation for future development and optimization.

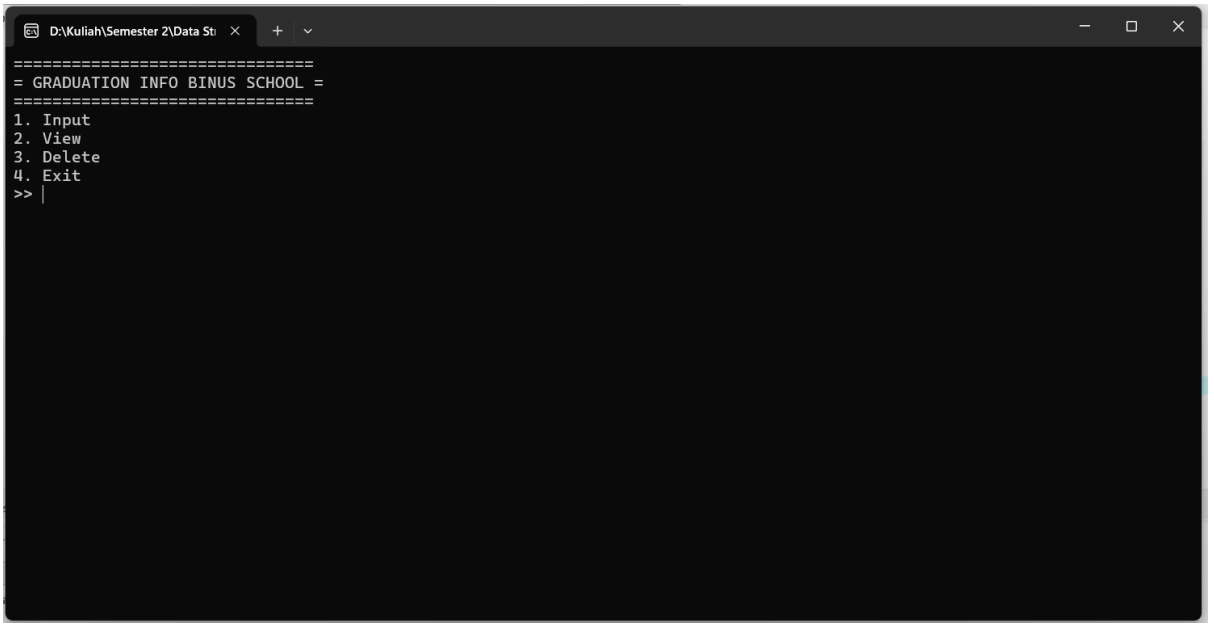
Time complexity compared to normal binary tree

	Worst Case Binary Tree	Worst Case AVL Tree
Insert Node	O(n)	O(log n)
Search Node	O(n)	O(log n)
Delete Node	O(n)	O(log n)

Number 4 :

All feature screenshoot

A. Main Menu



B. Insert Data Menu

1. Insert Yongky Alexander Tristan

```
D:\Kuliah\Semester 2\Data Sti
================================
= GRADUATION INFO BINUS SCHOOL =
================================
1. Input
2. View
3. Delete
4. Exit
>> 1
Input Name [5 s.d 50]: Yongky Alexander Tristan
Input your birth date [dd/mm/yyyy]: 16/07/2005
Enter your class [10-12 + A-L] (10B || 11L || 12F): 10B
Enter your major [IPA | IPS]: IPA
Enter your graduation year [2006-2024]: 2022
Insert Successfully!
Press any key to continue . . .
```

2. Insert Jose Antonio Marco

```
D:\Kuliah\Semester 2\Data Sti
================================
= GRADUATION INFO BINUS SCHOOL =
================================
1. Input
2. View
3. Delete
4. Exit
>> 1
Input Name [5 s.d 50]: Jose Antonio Marco
Input your birth date [dd/mm/yyyy]: 11/11/2011
Enter your class [10-12 + A-L] (10B || 11L || 12F): 12H
Enter your major [IPA | IPS]: IPS
Enter your graduation year [2006-2024]: 2018
Insert Successfully!
Press any key to continue . . .
```

3. Insert Muhammad Bachtiar Rasyid

```
D:\Kuliah\Semester 2\Data Stı
==============================
= GRADUATION INFO BINUS SCHOOL =
==============================
1. Input
2. View
3. Delete
4. Exit
>> 1
Input Name [5 s.d 50]: Muhammad Bachtiar Rasyid
Input your birth date [dd/mm/yyyy]: 22/12/2002
Enter your class [10-12 + A-L] (10B || 11L || 12F): 12L
Enter your major [IPA | IPS]: IPA
Enter your graduation year [2006-2024]: 2018
Insert Successfully!
Press any key to continue . . .
```

4. Insert Darren Nathanael Bhekti

```
D:\Kuliah\Semester 2\Data Stı
==============================
= GRADUATION INFO BINUS SCHOOL =
==============================
1. Input
2. View
3. Delete
4. Exit
>> 1
Input Name [5 s.d 50]: Darren Nathanael Bhekti
Input your birth date [dd/mm/yyyy]: 03/03/2003
Enter your class [10-12 + A-L] (10B || 11L || 12F): 11C
Enter your major [IPA | IPS]: IPS
Enter your graduation year [2006-2024]: 2019
Insert Successfully!
Press any key to continue . . .
```

5. Insert Jonathan Andrew Pratama

```
D:\Kuliah\Semester 2\Data St
================================
= GRADUATION INFO BINUS SCHOOL =
================================
1. Input
2. View
3. Delete
4. Exit
>> 1
Input Name [5 s.d 50]: Jonathan Andrew Pratama
Input your birth date [dd/mm/yyyy]: 04/04/2004
Enter your class [10-12 + A-L] (10B || 11L || 12F): 10G
Enter your major [IPA | IPS]: IPS
Enter your graduation year [2006-2024]: 2020
Insert Successfully!
Press any key to continue . . .
```

6. Insert Stevan Pohan

```
D:\Kuliah\Semester 2\Data St
================================
= GRADUATION INFO BINUS SCHOOL =
================================
1. Input
2. View
3. Delete
4. Exit
>> 1
Input Name [5 s.d 50]: Stevan Pohan
Input your birth date [dd/mm/yyyy]: 05/05/2005
Enter your class [10-12 + A-L] (10B || 11L || 12F): 12D
Enter your major [IPA | IPS]: IPA
Enter your graduation year [2006-2024]: 2016
Insert Successfully!
Press any key to continue . . .
```

7. Insert Marco Bennedict Makin

```
D:\Kuliah\Semester 2\Data St
================================
= GRADUATION INFO BINUS SCHOOL =
================================
1. Input
2. View
3. Delete
4. Exit
>> 1
Input Name [5 s.d 50]: Marco Bennedict Makin
Input your birth date [dd/mm/yyyy]: 06/06/2006
Enter your class [10-12 + A-L] (10B || 11L || 12F): 10A
Enter your major [IPA | IPS]: IPS
Enter your graduation year [2006-2024]: 2022
Insert Successfully!
Press any key to continue . . .
```

8. Insert James Dawson Haryanto

```
D:\Kuliah\Semester 2\Data St
================================
= GRADUATION INFO BINUS SCHOOL =
================================
1. Input
2. View
3. Delete
4. Exit
>> 1
Input Name [5 s.d 50]: James Dawson Haryanto
Input your birth date [dd/mm/yyyy]: 07/07/2007
Enter your class [10-12 + A-L] (10B || 11L || 12F): 12J
Enter your major [IPA | IPS]: IPA
Enter your graduation year [2006-2024]: 2017
Insert Successfully!
Press any key to continue . . .
```

9. Insert Brian Alexander

```
D:\Kuliah\Semester 2\Data St
==============================
= GRADUATION INFO BINUS SCHOOL =
==============================
1. Input
2. View
3. Delete
4. Exit
>> 1
Input Name [5 s.d 50]: Brian Alexander
Input your birth date [dd/mm/yyyy]: 08/08/2008
Enter your class [10-12 + A-L] (10B || 11L || 12F): 11F
Enter your major [IPA | IPS]: IPA
Enter your graduation year [2006-2024]: 2020
Insert Successfully!
Press any key to continue . . .
```

10. Insert Sergio Winero

```
D:\Kuliah\Semester 2\Data St
==============================
= GRADUATION INFO BINUS SCHOOL =
==============================
1. Input
2. View
3. Delete
4. Exit
>> 1
Input Name [5 s.d 50]: Sergio Winero
Input your birth date [dd/mm/yyyy]: 11/11/2011
Enter your class [10-12 + A-L] (10B || 11L || 12F): 10I
Enter your major [IPA | IPS]: IPS
Enter your graduation year [2006-2024]: 2024
Insert Successfully!
Press any key to continue . . .
```

C. View and Search Menu

1. Menu View

```
D:\Kuliah\Semester 2\Data Stu

================================
= GRADUATION INFO BINUS SCHOOL =
================================
Input view graduation :
1. View Preorder
2. View Inorder
3. View Postorder
4. View Certain Data
5. View Spesific (Search)
>>
```

2. View Preorder

```
D:\Kuliah\Semester 2\Data Stu

1. View Preorder
2. View Inorder
3. View Postorder
4. View Certain Data
5. View Spesific (Search)
>> 1
-------------------------------------------------------------------------------------------
| NISN       | Name                      | Birth Date | Class | Major | Graduation Year |
-------------------------------------------------------------------------------------------
| 2004001805 | Jonathan Andrew Pratama   | 04/04/2004 | 10G   | IPS   | 2020            |
-------------------------------------------------------------------------------------------
| 2003001554 | Darren Nathanael Bhekti   | 03/03/2003 | 11C   | IPS   | 2019            |
-------------------------------------------------------------------------------------------
| 2002001355 | Muhammad Bachtiar Rasyid  | 22/12/2002 | 12L   | IPA   | 2018            |
-------------------------------------------------------------------------------------------
| 2007002726 | James Dawson Haryanto     | 07/07/2007 | 12J   | IPA   | 2017            |
-------------------------------------------------------------------------------------------
| 2005032549 | Yongky Alexander Tristan  | 16/07/2005 | 10B   | IPA   | 2022            |
-------------------------------------------------------------------------------------------
| 2005002027 | Stevan Pohan              | 05/05/2005 | 12D   | IPA   | 2016            |
-------------------------------------------------------------------------------------------
| 2006002462 | Marco Bennedict Makin     | 06/06/2006 | 10A   | IPS   | 2022            |
-------------------------------------------------------------------------------------------
| 2011000241 | Jose Antonio Marco        | 11/11/2011 | 12H   | IPS   | 2018            |
-------------------------------------------------------------------------------------------
| 2008002886 | Brian Alexander           | 08/08/2008 | 11F   | IPA   | 2020            |
-------------------------------------------------------------------------------------------
| 2011003036 | Sergio Winero             | 11/11/2011 | 10I   | IPS   | 2024            |
-------------------------------------------------------------------------------------------
Press any key to continue . . .
```

3. View Inorder

```
D:\Kuliah\Semester 2\Data St

1. View Preorder
2. View Inorder
3. View Postorder
4. View Certain Data
5. View Spesific (Search)
>> 2
--------------------------------------------------------------------------------------------
| NISN       | Name                       | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------------------
| 2002001355 | Muhammad Bachtiar Rasyid   | 22/12/2002 | 12L   | IPA   | 2018            |
--------------------------------------------------------------------------------------------
| 2003001554 | Darren Nathanael Bhekti    | 03/03/2003 | 11C   | IPS   | 2019            |
--------------------------------------------------------------------------------------------
| 2004001805 | Jonathan Andrew Pratama    | 04/04/2004 | 10G   | IPS   | 2020            |
--------------------------------------------------------------------------------------------
| 2005002027 | Stevan Pohan               | 05/05/2005 | 12D   | IPA   | 2016            |
--------------------------------------------------------------------------------------------
| 2005032549 | Yongky Alexander Tristan   | 16/07/2005 | 10B   | IPA   | 2022            |
--------------------------------------------------------------------------------------------
| 2006002462 | Marco Bennedict Makin      | 06/06/2006 | 10A   | IPS   | 2022            |
--------------------------------------------------------------------------------------------
| 2007002726 | James Dawson Haryanto      | 07/07/2007 | 12J   | IPA   | 2017            |
--------------------------------------------------------------------------------------------
| 2008002886 | Brian Alexander            | 08/08/2008 | 11F   | IPA   | 2020            |
--------------------------------------------------------------------------------------------
| 2011000241 | Jose Antonio Marco         | 11/11/2011 | 12H   | IPS   | 2018            |
--------------------------------------------------------------------------------------------
| 2011003036 | Sergio Winero              | 11/11/2011 | 10I   | IPS   | 2024            |
--------------------------------------------------------------------------------------------
Press any key to continue . . .
```

4. View Postorder

```
D:\Kuliah\Semester 2\Data St

1. View Preorder
2. View Inorder
3. View Postorder
4. View Certain Data
5. View Spesific (Search)
>> 3
--------------------------------------------------------------------------------------------
| NISN       | Name                       | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------------------
| 2002001355 | Muhammad Bachtiar Rasyid   | 22/12/2002 | 12L   | IPA   | 2018            |
--------------------------------------------------------------------------------------------
| 2003001554 | Darren Nathanael Bhekti    | 03/03/2003 | 11C   | IPS   | 2019            |
--------------------------------------------------------------------------------------------
| 2005002027 | Stevan Pohan               | 05/05/2005 | 12D   | IPA   | 2016            |
--------------------------------------------------------------------------------------------
| 2006002462 | Marco Bennedict Makin      | 06/06/2006 | 10A   | IPS   | 2022            |
--------------------------------------------------------------------------------------------
| 2005032549 | Yongky Alexander Tristan   | 16/07/2005 | 10B   | IPA   | 2022            |
--------------------------------------------------------------------------------------------
| 2008002886 | Brian Alexander            | 08/08/2008 | 11F   | IPA   | 2020            |
--------------------------------------------------------------------------------------------
| 2011003036 | Sergio Winero              | 11/11/2011 | 10I   | IPS   | 2024            |
--------------------------------------------------------------------------------------------
| 2011000241 | Jose Antonio Marco         | 11/11/2011 | 12H   | IPS   | 2018            |
--------------------------------------------------------------------------------------------
| 2007002726 | James Dawson Haryanto      | 07/07/2007 | 12J   | IPA   | 2017            |
--------------------------------------------------------------------------------------------
| 2004001805 | Jonathan Andrew Pratama    | 04/04/2004 | 10G   | IPS   | 2020            |
--------------------------------------------------------------------------------------------
Press any key to continue . . .
```

5. View Certain “IPA” Data

```
D:\Kuliah\Semester 2\Data St
5. View Spesific (Search)
>> 4
Enter major to search [IPA | IPS]: IPA
Students with major IPA:
--------------------------------------------------------------------------------
| NISN       | Name                       | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------
| 2002001355 | Muhammad Bachtiar Rasyid   | 22/12/2002 | 12L   | IPA   | 2018            |
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
| NISN       | Name                       | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------
| 2005002027 | Stevan Pohan               | 05/05/2005 | 12D   | IPA   | 2016            |
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
| NISN       | Name                       | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------
| 2005032549 | Yongky Alexander Tristan   | 16/07/2005 | 10B   | IPA   | 2022            |
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
| NISN       | Name                       | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------
| 2007002726 | James Dawson Haryanto      | 07/07/2007 | 12J   | IPA   | 2017            |
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
| NISN       | Name                       | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------
| 2008002886 | Brian Alexander            | 08/08/2008 | 11F   | IPA   | 2020            |
--------------------------------------------------------------------------------
Press any key to continue . . .
```

6. View Certain “IPS” Data

```
D:\Kuliah\Semester 2\Data St
5. View Spesific (Search)
>> 4
Enter major to search [IPA | IPS]: IPS
Students with major IPS:
--------------------------------------------------------------------------------
| NISN       | Name                       | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------
| 2003001554 | Darren Nathanael Bhekti    | 03/03/2003 | 11C   | IPS   | 2019            |
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
| NISN       | Name                       | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------
| 2004001805 | Jonathan Andrew Pratama    | 04/04/2004 | 10G   | IPS   | 2020            |
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
| NISN       | Name                       | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------
| 2006002462 | Marco Bennedict Makin      | 06/06/2006 | 10A   | IPS   | 2022            |
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
| NISN       | Name                       | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------
| 2011000241 | Jose Antonio Marco         | 11/11/2011 | 12H   | IPS   | 2018            |
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
| NISN       | Name                       | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------
| 2011003036 | Sergio Winero              | 11/11/2011 | 10I   | IPS   | 2024            |
--------------------------------------------------------------------------------
Press any key to continue . . .
```

7. View Spesific (Search) Awal

```
D:\Kuliah\Semester 2\Data St
1. View Preorder
2. View Inorder
3. View Postorder
4. View Certain Data
5. View Spesific (Search)
>> 5
--------------------------------------------------------------------------------------
| NISN       | Name                       | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------------
| 2002001355 | Muhammad Bachtiar Rasyid   | 22/12/2002 | 12L   | IPA   | 2018            |
--------------------------------------------------------------------------------------
| 2003001554 | Darren Nathanael Bhekti    | 03/03/2003 | 11C   | IPS   | 2019            |
--------------------------------------------------------------------------------------
| 2004001805 | Jonathan Andrew Pratama    | 04/04/2004 | 10G   | IPS   | 2020            |
--------------------------------------------------------------------------------------
| 2005002027 | Stevan Pohan               | 05/05/2005 | 12D   | IPA   | 2016            |
--------------------------------------------------------------------------------------
| 2005032549 | Yongky Alexander Tristan   | 16/07/2005 | 10B   | IPA   | 2022            |
--------------------------------------------------------------------------------------
| 2006002462 | Marco Bennedict Makin      | 06/06/2006 | 10A   | IPS   | 2022            |
--------------------------------------------------------------------------------------
| 2007002726 | James Dawson Haryanto      | 07/07/2007 | 12J   | IPA   | 2017            |
--------------------------------------------------------------------------------------
| 2008002886 | Brian Alexander            | 08/08/2008 | 11F   | IPA   | 2020            |
--------------------------------------------------------------------------------------
| 2011000241 | Jose Antonio Marco         | 11/11/2011 | 12H   | IPS   | 2018            |
--------------------------------------------------------------------------------------
| 2011003036 | Sergio Winero              | 11/11/2011 | 10I   | IPS   | 2024            |
--------------------------------------------------------------------------------------
Enter the 10-digit NISN:
```

8. View Search “2007002726” a/n James Dawson Haryanto

```
D:\Kuliah\Semester 2\Data St
--------------------------------------------------------------------------------------
| NISN       | Name                       | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------------
| 2002001355 | Muhammad Bachtiar Rasyid   | 22/12/2002 | 12L   | IPA   | 2018            |
--------------------------------------------------------------------------------------
| 2003001554 | Darren Nathanael Bhekti    | 03/03/2003 | 11C   | IPS   | 2019            |
--------------------------------------------------------------------------------------
| 2004001805 | Jonathan Andrew Pratama    | 04/04/2004 | 10G   | IPS   | 2020            |
--------------------------------------------------------------------------------------
| 2005002027 | Stevan Pohan               | 05/05/2005 | 12D   | IPA   | 2016            |
--------------------------------------------------------------------------------------
| 2005032549 | Yongky Alexander Tristan   | 16/07/2005 | 10B   | IPA   | 2022            |
--------------------------------------------------------------------------------------
| 2006002462 | Marco Bennedict Makin      | 06/06/2006 | 10A   | IPS   | 2022            |
--------------------------------------------------------------------------------------
| 2007002726 | James Dawson Haryanto      | 07/07/2007 | 12J   | IPA   | 2017            |
--------------------------------------------------------------------------------------
| 2008002886 | Brian Alexander            | 08/08/2008 | 11F   | IPA   | 2020            |
--------------------------------------------------------------------------------------
| 2011000241 | Jose Antonio Marco         | 11/11/2011 | 12H   | IPS   | 2018            |
--------------------------------------------------------------------------------------
| 2011003036 | Sergio Winero              | 11/11/2011 | 10I   | IPS   | 2024            |
--------------------------------------------------------------------------------------
Enter the 10-digit NISN: 2007002726
--------------------------------------------------------------------------------------
| NISN       | Name                       | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------------
| 2007002726 | James Dawson Haryanto      | 07/07/2007 | 12J   | IPA   | 2017            |
--------------------------------------------------------------------------------------
Press any key to continue . . .
```

9. View Search “2005032549” a/n Yongky Alexander Tristan

```
D:\Kuliah\Semester 2\Data St
--------------------------------------------------------------------------------
| NISN       | Name                      | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------
| 2002001355 | Muhammad Bachtiar Rasyid  | 22/12/2002 | 12L   | IPA   | 2018            |
--------------------------------------------------------------------------------
| 2003001554 | Darren Nathanael Bhekti   | 03/03/2003 | 11C   | IPS   | 2019            |
--------------------------------------------------------------------------------
| 2004001805 | Jonathan Andrew Pratama   | 04/04/2004 | 10G   | IPS   | 2020            |
--------------------------------------------------------------------------------
| 2005002027 | Stevan Pohan              | 05/05/2005 | 12D   | IPA   | 2016            |
--------------------------------------------------------------------------------
| 2005032549 | Yongky Alexander Tristan  | 16/07/2005 | 10B   | IPA   | 2022            |
--------------------------------------------------------------------------------
| 2006002462 | Marco Bennedict Makin     | 06/06/2006 | 10A   | IPS   | 2022            |
--------------------------------------------------------------------------------
| 2007002726 | James Dawson Haryanto     | 07/07/2007 | 12J   | IPA   | 2017            |
--------------------------------------------------------------------------------
| 2008002886 | Brian Alexander           | 08/08/2008 | 11F   | IPA   | 2020            |
--------------------------------------------------------------------------------
| 2011000241 | Jose Antonio Marco        | 11/11/2011 | 12H   | IPS   | 2018            |
--------------------------------------------------------------------------------
| 2011003036 | Sergio Winero             | 11/11/2011 | 10I   | IPS   | 2024            |
--------------------------------------------------------------------------------
Enter the 10-digit NISN: 2005032549
--------------------------------------------------------------------------------
| NISN       | Name                      | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------
| 2005032549 | Yongky Alexander Tristan  | 16/07/2005 | 10B   | IPA   | 2022            |
--------------------------------------------------------------------------------
Press any key to continue . . .
```

10. View Search “2005002027” a/n Stevan Pohan

```
D:\Kuliah\Semester 2\Data St
--------------------------------------------------------------------------------
| NISN       | Name                      | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------
| 2002001355 | Muhammad Bachtiar Rasyid  | 22/12/2002 | 12L   | IPA   | 2018            |
--------------------------------------------------------------------------------
| 2003001554 | Darren Nathanael Bhekti   | 03/03/2003 | 11C   | IPS   | 2019            |
--------------------------------------------------------------------------------
| 2004001805 | Jonathan Andrew Pratama   | 04/04/2004 | 10G   | IPS   | 2020            |
--------------------------------------------------------------------------------
| 2005002027 | Stevan Pohan              | 05/05/2005 | 12D   | IPA   | 2016            |
--------------------------------------------------------------------------------
| 2005032549 | Yongky Alexander Tristan  | 16/07/2005 | 10B   | IPA   | 2022            |
--------------------------------------------------------------------------------
| 2006002462 | Marco Bennedict Makin     | 06/06/2006 | 10A   | IPS   | 2022            |
--------------------------------------------------------------------------------
| 2007002726 | James Dawson Haryanto     | 07/07/2007 | 12J   | IPA   | 2017            |
--------------------------------------------------------------------------------
| 2008002886 | Brian Alexander           | 08/08/2008 | 11F   | IPA   | 2020            |
--------------------------------------------------------------------------------
| 2011000241 | Jose Antonio Marco        | 11/11/2011 | 12H   | IPS   | 2018            |
--------------------------------------------------------------------------------
| 2011003036 | Sergio Winero             | 11/11/2011 | 10I   | IPS   | 2024            |
--------------------------------------------------------------------------------
Enter the 10-digit NISN: 2005002027
--------------------------------------------------------------------------------
| NISN       | Name                      | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------
| 2005002027 | Stevan Pohan              | 05/05/2005 | 12D   | IPA   | 2016            |
--------------------------------------------------------------------------------
Press any key to continue . . .
```

D. Delete Menu

1. Delete Menu Awal

```
D:\Kuliah\Semester 2\Data St

================================
1. Input
2. View
3. Delete
4. Exit
>> 3
--------------------------------------------------------------------------------------
| NISN       | Name                       | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------------
| 2002001355 | Muhammad Bachtiar Rasyid   | 22/12/2002 | 12L   | IPA   | 2018            |
--------------------------------------------------------------------------------------
| 2003001554 | Darren Nathanael Bhekti    | 03/03/2003 | 11C   | IPS   | 2019            |
--------------------------------------------------------------------------------------
| 2004001805 | Jonathan Andrew Pratama    | 04/04/2004 | 10G   | IPS   | 2020            |
--------------------------------------------------------------------------------------
| 2005002027 | Stevan Pohan               | 05/05/2005 | 12D   | IPA   | 2016            |
--------------------------------------------------------------------------------------
| 2005032549 | Yongky Alexander Tristan   | 16/07/2005 | 10B   | IPA   | 2022            |
--------------------------------------------------------------------------------------
| 2006002462 | Marco Bennedict Makin      | 06/06/2006 | 10A   | IPS   | 2022            |
--------------------------------------------------------------------------------------
| 2007002726 | James Dawson Haryanto      | 07/07/2007 | 12J   | IPA   | 2017            |
--------------------------------------------------------------------------------------
| 2008002886 | Brian Alexander            | 08/08/2008 | 11F   | IPA   | 2020            |
--------------------------------------------------------------------------------------
| 2011000241 | Jose Antonio Marco         | 11/11/2011 | 12H   | IPS   | 2018            |
--------------------------------------------------------------------------------------
| 2011003036 | Sergio Winero              | 11/11/2011 | 10I   | IPS   | 2024            |
--------------------------------------------------------------------------------------
Input NISN to Delete:
```

2. Delete “2004001805” a/n Jonathan Andrew Pratama

```
D:\Kuliah\Semester 2\Data St

2. View
3. Delete
4. Exit
>> 3
--------------------------------------------------------------------------------------
| NISN       | Name                       | Birth Date | Class | Major | Graduation Year |
--------------------------------------------------------------------------------------
| 2002001355 | Muhammad Bachtiar Rasyid   | 22/12/2002 | 12L   | IPA   | 2018            |
--------------------------------------------------------------------------------------
| 2003001554 | Darren Nathanael Bhekti    | 03/03/2003 | 11C   | IPS   | 2019            |
--------------------------------------------------------------------------------------
| 2004001805 | Jonathan Andrew Pratama    | 04/04/2004 | 10G   | IPS   | 2020            |
--------------------------------------------------------------------------------------
| 2005002027 | Stevan Pohan               | 05/05/2005 | 12D   | IPA   | 2016            |
--------------------------------------------------------------------------------------
| 2005032549 | Yongky Alexander Tristan   | 16/07/2005 | 10B   | IPA   | 2022            |
--------------------------------------------------------------------------------------
| 2006002462 | Marco Bennedict Makin      | 06/06/2006 | 10A   | IPS   | 2022            |
--------------------------------------------------------------------------------------
| 2007002726 | James Dawson Haryanto      | 07/07/2007 | 12J   | IPA   | 2017            |
--------------------------------------------------------------------------------------
| 2008002886 | Brian Alexander            | 08/08/2008 | 11F   | IPA   | 2020            |
--------------------------------------------------------------------------------------
| 2011000241 | Jose Antonio Marco         | 11/11/2011 | 12H   | IPS   | 2018            |
--------------------------------------------------------------------------------------
| 2011003036 | Sergio Winero              | 11/11/2011 | 10I   | IPS   | 2024            |
--------------------------------------------------------------------------------------
Input NISN to Delete: 2004001805
Delete successfully
Press any key to continue . . .
```

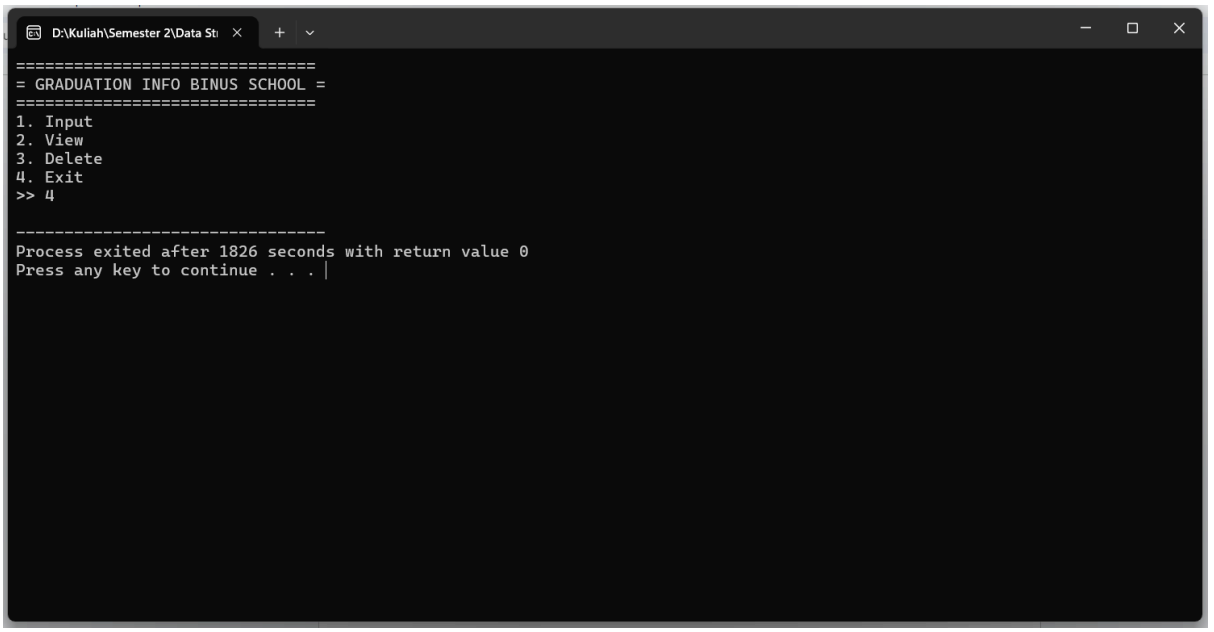
3. Delete “2008002886” a/n Brian Alexander

```
D:\Kuliah\Semester 2\Data St
==============================
1. Input
2. View
3. Delete
4. Exit
>> 3
-------------------------------------------------------------------------------------------
| NISN       | Name                       | Birth Date | Class | Major | Graduation Year |
-------------------------------------------------------------------------------------------
| 2002001355 | Muhammad Bachtiar Rasyid   | 22/12/2002 | 12L   | IPA   | 2018            |
-------------------------------------------------------------------------------------------
| 2003001554 | Darren Nathanael Bhekti    | 03/03/2003 | 11C   | IPS   | 2019            |
-------------------------------------------------------------------------------------------
| 2005002027 | Stevan Pohan               | 05/05/2005 | 12D   | IPA   | 2016            |
-------------------------------------------------------------------------------------------
| 2005032549 | Yongky Alexander Tristan   | 16/07/2005 | 10B   | IPA   | 2022            |
-------------------------------------------------------------------------------------------
| 2006002462 | Marco Bennedict Makin      | 06/06/2006 | 10A   | IPS   | 2022            |
-------------------------------------------------------------------------------------------
| 2007002726 | James Dawson Haryanto      | 07/07/2007 | 12J   | IPA   | 2017            |
-------------------------------------------------------------------------------------------
| 2008002886 | Brian Alexander            | 08/08/2008 | 11F   | IPA   | 2020            |
-------------------------------------------------------------------------------------------
| 2011000241 | Jose Antonio Marco         | 11/11/2011 | 12H   | IPS   | 2018            |
-------------------------------------------------------------------------------------------
| 2011003036 | Sergio Winero              | 11/11/2011 | 10I   | IPS   | 2024            |
-------------------------------------------------------------------------------------------
Input NISN to Delete: 2008002886
Delete successfully
Press any key to continue . . .
```

4. Delete Random “2702254676”

```
D:\Kuliah\Semester 2\Data St
==============================
= GRADUATION INFO BINUS SCHOOL =
==============================
1. Input
2. View
3. Delete
4. Exit
>> 3
-------------------------------------------------------------------------------------------
| NISN       | Name                       | Birth Date | Class | Major | Graduation Year |
-------------------------------------------------------------------------------------------
| 2002001355 | Muhammad Bachtiar Rasyid   | 22/12/2002 | 12L   | IPA   | 2018            |
-------------------------------------------------------------------------------------------
| 2003001554 | Darren Nathanael Bhekti    | 03/03/2003 | 11C   | IPS   | 2019            |
-------------------------------------------------------------------------------------------
| 2005002027 | Stevan Pohan               | 05/05/2005 | 12D   | IPA   | 2016            |
-------------------------------------------------------------------------------------------
| 2005032549 | Yongky Alexander Tristan   | 16/07/2005 | 10B   | IPA   | 2022            |
-------------------------------------------------------------------------------------------
| 2006002462 | Marco Bennedict Makin      | 06/06/2006 | 10A   | IPS   | 2022            |
-------------------------------------------------------------------------------------------
| 2007002726 | James Dawson Haryanto      | 07/07/2007 | 12J   | IPA   | 2017            |
-------------------------------------------------------------------------------------------
| 2011000241 | Jose Antonio Marco         | 11/11/2011 | 12H   | IPS   | 2018            |
-------------------------------------------------------------------------------------------
| 2011003036 | Sergio Winero              | 11/11/2011 | 10I   | IPS   | 2024            |
-------------------------------------------------------------------------------------------
Input NISN to Delete: 2702254676
Data not found
Press any key to continue . . .
```

E. Exit



Program Code :

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <time.h>
#include <stdbool.h>

struct Node {
    char name[51];
    char classes[21];
    char jurusan[21];
    int nisn;
    char born[41];
    int grad;

    int height;
    struct Node* left;
    struct Node* right;
}*root = NULL;

bool deleteFound = false;

struct Node* createNode(char name[], char classes[], char jurusan[], int nisn, char born[], int
grad) {
    struct Node* newNode = (struct Node*) malloc (sizeof(struct Node));
    strcpy(newNode->name, name);
```

```
    strcpy(newNode->classes, classes);
    strcpy(newNode->jurusan, jurusan);
    strcpy(newNode->born, born);
    newNode->nisn = nisn;
    newNode->grad = grad;
    newNode->left = NULL;
    newNode->right = NULL;
    newNode->height = 1;

    return newNode;
}

int max(int a, int b) {
    return (a > b) ? a : b;
}

int getHeight(struct Node* node) {
    if(node == NULL) return 0;
    return node->height;
}

int setHeight(struct Node* node) {
    return max(getHeight(node->left), getHeight(node->right)) + 1;
}

int getBalanceFactor(struct Node* node) {
    return getHeight(node->left) - getHeight(node->right);
}

struct Node* rotateLeft(struct Node* x) {
    struct Node* y = x->right;
    struct Node* z = y->left;

    x->right = z;
    y->left = x;

    x->height = setHeight(x);
    y->height = setHeight(y);

    return y;
}

struct Node* rotateRight(struct Node* x) {
    struct Node* y = x->left;
```

```
    struct Node* z = y->right;

    x->left = z;
    y->right = x;

    x->height = setHeight(x);
    y->height = setHeight(y);

    return y;
}

struct Node* insertNode(struct Node* curr, char name[], char classes[], char jurusan[], char
born[], int nisn, int grad) {
    if (curr == NULL) return createNode(name, classes, jurusan, nisn, born, grad);

    if (nisn < curr->nisn) {
        curr->left = insertNode(curr->left, name, classes, jurusan, born, nisn, grad);
    } else if (nisn > curr->nisn) {
        curr->right = insertNode(curr->right, name, classes, jurusan, born, nisn, grad);
    } else {
        return curr;
    }

    curr->height = setHeight(curr);
    int bf = getBalanceFactor(curr);

    if (bf > 1) {
        if (nisn < curr->left->nisn) return rotateRight(curr);
        else {
            curr->left = rotateLeft(curr->left);
            return rotateRight(curr);
        }
    } else if (bf < -1) {
        if (nisn > curr->right->nisn) return rotateLeft(curr);
        else {
            curr->right = rotateRight(curr->right);
            return rotateLeft(curr);
        }
    }
    return curr;
}

struct Node* deleteNode(struct Node* curr, int nisn) {
    if(curr == NULL) return NULL;
```

```
if(nisn < curr->nisn) {
    curr->left = deleteNode(curr->left, nisn);
} else if(nisn > curr->nisn) {
    curr->right = deleteNode(curr->right, nisn);
} else {
    deleteFound = true;
    if(curr->left == NULL && curr->right == NULL) {
        free(curr);
        return NULL;
    }
    if(curr->left == NULL) {
        struct Node* temp = curr->right;
        free(curr);
        return temp;
    } else if(curr->right == NULL) {
        struct Node* temp = curr->left;
        free(curr);
        return temp;
    }

    struct Node* is = curr->right;
    while(is->left != NULL) {
        is = is->left;
    }

    strcpy(curr->name, is->name);
    strcpy(curr->classes, is->classes);
    strcpy(curr->jurusan, is->jurusan);
    curr->nisn = is->nisn;
    strcpy(curr->born, is->born);
    curr->grad = is->grad;

    curr->right = deleteNode(curr->right, is->nisn);
}

curr->height = setHeight(curr);

int bf = getBalanceFactor(curr);

if(bf > 1) {
    if(getBalanceFactor(curr->left) >= 0) curr = rotateRight(curr);
    else {
        curr->left = rotateLeft(curr->left);
```

```
            curr = rotateRight(curr);
        }
    } else if (bf < -1) {
        if(getBalanceFactor(curr->right) <= 0) curr = rotateLeft(curr);
        else {
            curr->right = rotateRight(curr->right);
            curr = rotateLeft(curr);
        }
    }

    return curr;
}

void printRow(int nisn, char name[], char born[], char classes[], char jurusan[], int grad) {
    printf("| %d | %-25s | %-10s | %-5s | %-5s | %-15d |\n", nisn, name, born, classes, jurusan,
grad);
    printf("-----------------------------------------------------------------------------------------\n");
}

void viewMajor(struct Node* curr, char jurusan[], int *countMajor) {
    if (curr == NULL) return;

    viewMajor(curr->left, jurusan, countMajor);

    if (strcmp(curr->jurusan, jurusan) == 0) {
            printf("-----------------------------------------------------------------------------------------\n");
                    printf("| NISN      | Name                          | Birth Date | Class | Major |
Graduation Year |\n");

printf("-----------------------------------------------------------------------------------------\n");
        printRow(curr->nisn, curr->name, curr->born, curr->classes, curr->jurusan, curr->grad);
        (*countMajor)++;
    }

    viewMajor(curr->right, jurusan, countMajor);
}

void searchNISN(struct Node* curr, int nisn, int *countNISN) {
    if (curr == NULL) return;

    searchNISN(curr->left, nisn, countNISN);

    if (curr->nisn == nisn) {
            printf("-----------------------------------------------------------------------------------------\n");
```

```
                printf("| NISN      | Name                          | Birth Date | Class | Major |
Graduation Year |\n");

printf("-----------------------------------------------------------------------------------------\n");
        printRow(curr->nisn, curr->name, curr->born, curr->classes, curr->jurusan, curr->grad);
        (*countNISN)++;
    }

    searchNISN(curr->right, nisn, countNISN);
}

void viewData(struct Node* curr, int choose, bool *headerPrinted) {
    if (root == NULL) {
        printf("No data available!\n");
        return;
    }

            if (!(*headerPrinted) && (choose == 1 || choose == 2 || choose == 3)) {
        printf("-----------------------------------------------------------------------------------------\n");
                printf("| NISN      | Name                          | Birth Date | Class | Major |
Graduation Year |\n");

printf("-----------------------------------------------------------------------------------------\n");
        *headerPrinted = true;
    }

    if (curr == NULL) return;

    if (choose == 1) { // preorder
        printRow(curr->nisn, curr->name, curr->born, curr->classes, curr->jurusan, curr->grad);
        viewData(curr->left, choose, headerPrinted);
        viewData(curr->right, choose, headerPrinted);
    } else if (choose == 2) { // inorder
        viewData(curr->left, choose, headerPrinted);
        printRow(curr->nisn, curr->name, curr->born, curr->classes, curr->jurusan, curr->grad);
        viewData(curr->right, choose, headerPrinted);
    } else if (choose == 3) { // postorder
        viewData(curr->left, choose, headerPrinted);
        viewData(curr->right, choose, headerPrinted);
        printRow(curr->nisn, curr->name, curr->born, curr->classes, curr->jurusan, curr->grad);
    } else if (choose == 4){
            char jurusan[21];
            int countMajor = 0;
```

```
        do{
            printf("Enter major to search [IPA | IPS]: ");
            gets(jurusan);

            if(strcmp(jurusan, "IPA") != 0 && strcmp(jurusan, "IPS") != 0) {
                printf("Invalid major entered. Please enter 'IPA' or 'IPS'\n");
                continue;
            }

            int countMajor = 0;
            printf("Students with major %s:\n", jurusan);
            viewMajor(root, jurusan, &countMajor);

            if (countMajor == 0) {
                printf("No students found with major %s.\n", jurusan);
            }
                    } while(strcmp(jurusan, "IPA") != 0 && strcmp(jurusan, "IPS") !=0);

            } else if (choose == 5){
                    int nisn;
        int countNISN = 0;
        viewData(root, 2, headerPrinted);
        printf("Enter the 10-digit NISN: ");
        scanf("%10d", &nisn); getchar();


        searchNISN(root, nisn, &countNISN);

        if (countNISN == 0) {
            printf("No students found with NISN %d.\n", nisn);
        }
            }

}

bool validateClasses(char classes[]) {
    int length = strlen(classes);

    if (length != 3) return false;

    if (classes[0] != '1') return false;
    if (classes[1] < '0' || classes[1] > '2') return false;

    char huruf = classes[length - 1];
```

```
    if (huruf < 'A' || huruf > 'L') return false;

    return true;
}

bool validateBorn(char born[]) {
    if (strlen(born) != 10) return false;

    if (born[2] != '/' || born[5] != '/') return false;

    int day = (born[0] - '0') * 10 + (born[1] - '0');
    int month = (born[3] - '0') * 10 + (born[4] - '0');
    int year = (born[6] - '0') * 1000 + (born[7] - '0') * 100 + (born[8] - '0') * 10 + (born[9] - '0');

    if (day < 1 || day > 30) return false;
    if (month < 1 || month > 12) return false;
    if (year < 1990) return false;

    return true;
}

void insertMenu() {
    char name[51];
    char classes[21];
    char jurusan[21];
    int nisn;
    char born[41];
    int grad;

    do {
        printf("Input Name [5 s.d 50]: ");
        fgets(name, sizeof(name), stdin);
        name[strcspn(name, "\n")] = 0;
    } while (strlen(name) < 5 || strlen(name) > 50);

    do {
        printf("Input your birth date [dd/mm/yyyy]: ");
        fgets(born, sizeof(born), stdin);
        born[strcspn(born, "\n")] = 0;
    } while (!validateBorn(born));

    do {
        printf("Enter your class [10-12 + A-L] (10B || 11L || 12F): ");
        fgets(classes, sizeof(classes), stdin);
```

```
        classes[strcspn(classes, "\n")] = 0;
    } while (!validateClasses(classes));

    do {
        printf("Enter your major [IPA | IPS]: ");
        fgets(jurusan, sizeof(jurusan), stdin);
        jurusan[strcspn(jurusan, "\n")] = 0;
    } while (strcmp(jurusan, "IPA") != 0 && strcmp(jurusan, "IPS") != 0);

    do {
        printf("Enter your graduation year [2006-2024]: ");
        scanf("%d", &grad); getchar();
    } while (grad < 2006 || grad > 2024);

    char yearStr[5];
    strncpy(yearStr, born + 6, 4);
    yearStr[4] = '\0';
    srand(time(NULL));
    int random = rand() % 1000000;
    char nisnStr[11];
    sprintf(nisnStr, "%s%06d", yearStr, random);
    nisn = atoi(nisnStr);

    root = insertNode(root, name, classes, jurusan, born, nisn, grad);
    printf("Insert Successfully!\n");
    system("pause");
}

void deleteMenu() {
            bool headerPrinted = false;
            viewData(root, 2, &headerPrinted);
    if(root == NULL) return;

    int nisn;
    printf("Input NISN to Delete: ");
    scanf("%d", &nisn); getchar();

    root = deleteNode(root, nisn);
    if(deleteFound) {
        deleteFound = false;
        printf("Delete successfully\n");
    } else {
        printf("Data not found\n");
    }
```

```
    system("pause");
}

int main() {
    int option = -1;
    while (option != 4) {
            system("cls");
        printf("=================================\n");
        printf("= GRADUATION INFO BINUS SCHOOL =\n");
        printf("=================================\n");
        printf("1. Input\n");
        printf("2. View\n");
        printf("3. Delete\n");
        printf("4. Exit\n");
        printf(">> ");
        scanf("%d", &option); getchar();

        switch (option) {
            case 1 : {
                insertMenu();
                break;
            }
            case 2 : {
                    if (root == NULL) {
                                    printf("No data available!\n");
                                    }
                    else {
                            bool headerPrinted = false;
                            int choose;

                            system ("cls");
                            printf("=================================\n");
                            printf("= GRADUATION INFO BINUS SCHOOL =\n");
                            printf("=================================\n");
                            printf("Input view graduation :\n");
                            printf("1. View Preorder\n");
                            printf("2. View Inorder\n");
                            printf("3. View Postorder\n");
                            printf("4. View Certain Data\n");
                            printf("5. View Spesific (Search)\n");
                            printf(">> ");
                            scanf("%d", &choose); getchar();
                            switch (choose){
                                    case 1 : {
```

```
                                        viewData(root, 1, &headerPrinted);
                                                break;
                                        }
                                        case 2 : {
                                                viewData(root, 2, &headerPrinted);
                                                break;
                                        }
                                        case 3 : {
                                                viewData(root, 3, &headerPrinted);
                                                break;
                                        }
                                        case 4 : {
                                                viewData(root, 4, &headerPrinted);
                                                break;
                                        }
                                        case 5 : {
                                                viewData(root, 5, &headerPrinted);
                                                break;
                                        }
                                        default : {
                                                printf("Input view 1-5 :\n");
                                                break;
                                        }
                                }
                        }
            system("pause");
            break;
        }
        case 3 : {
            deleteMenu();
            break;
        }
        case 4 : {
            exit(0);
            break;
        }
        default : {
            printf("Invalid Input!\n");
        }
    }
  }

  return 0;
}
```

CODING DOCUMENTATION (WITH HUMAN LANGUAGE)

You can check // for an explanation regarding the code

```c
#include <stdio.h>      //library yang digunakan
#include <string.h>     //library yang digunakan
#include <stdlib.h>     //library yang digunakan
#include <time.h>       //library yang digunakan
#include <stdbool.h>    //library yang digunakan

struct Node {           //Declare struct yang berisi inputan dan height yang digunakan program
    char name[51];
    char classes[21];
    char jurusan[21];
    int nisn;
    char born[41];
    int grad;

    int height;
    struct Node* left;  //karena ini avl tree jadi ada kanan kiri dan height
    struct Node* right;
}*root = NULL;

bool deleteFound = false; // Untuk cek ada yang bisa di delete atau tidak
```

```c
struct Node* createNode(char name[], char classes[], char jurusan[], int nisn, char born[], int grad) {
    struct Node* newNode = (struct Node*) malloc (sizeof(struct Node));
    strcpy(newNode->name, name);
    strcpy(newNode->classes, classes);
    strcpy(newNode->jurusan, jurusan);  //fuction ini untuk masukin data ke program
    strcpy(newNode->born, born);
    newNode->nisn = nisn;
    newNode->grad = grad;
    newNode->left = NULL;
    newNode->right = NULL;
    newNode->height = 1;

    return newNode;
}

int max(int a, int b) {     //untuk cari tau max nya dimana
    return (a > b) ? a : b;
}

int getHeight(struct Node* node) {  //untuk mendapatkan nilai tertinggi
    if(node == NULL) return 0;
    return node->height;
}
```

```c
int setHeight(struct Node* node) { //untuk menjadikan node tersebut adalah nilai tertinggi
    return max(getHeight(node->left), getHeight(node->right)) + 1;
}

int getBalanceFactor(struct Node* node) { //karna ini avl jadi ada balancing
    return getHeight(node->left) - getHeight(node->right);
}

struct Node* rotateLeft(struct Node* x) { //untuk balancing
    struct Node* y = x->right;
    struct Node* z = y->left;

    x->right = z;
    y->left = x;

    x->height = setHeight(x);
    y->height = setHeight(y);

    return y;
}
```

```
struct Node* rotateRight(struct Node* x) { //untuk balancing
    struct Node* y = x->left;
    struct Node* z = y->right;

    x->left = z;
    y->right = x;

    x->height = setHeight(x);
    y->height = setHeight(y);

    return y;
}

```

```
struct Node* insertNode(struct Node* curr, char name[], char classes[], char jurusan[], char born[], int nisn, int grad) {
    // fuction ini untuk insert dan langsung di set sebagai avl tree
    if (curr == NULL) return createNode(name, classes, jurusan, nisn, born, grad);

    if (nisn < curr->nisn) {
        curr->left = insertNode(curr->left, name, classes, jurusan, born, nisn, grad);
    } else if (nisn > curr->nisn) {
        curr->right = insertNode(curr->right, name, classes, jurusan, born, nisn, grad);
    } else {
        return curr;
    }

    curr->height = setHeight(curr);
    int bf = getBalanceFactor(curr);

    if (bf > 1) {
        if (nisn < curr->left->nisn) return rotateRight(curr);
        else {
            curr->left = rotateLeft(curr->left);
            return rotateRight(curr);
        }
    } else if (bf < -1) {
        if (nisn > curr->right->nisn) return rotateLeft(curr);
        else {
            curr->right = rotateRight(curr->right);
            return rotateLeft(curr);
        }
    }
    return curr;
}

```

```
struct Node* deleteNode(struct Node* curr, int nisn) {
    // function ini untuk delete dan langsung balancing sebagai avl
    if(curr == NULL) return NULL;

    if(nisn < curr->nisn) {
        curr->left = deleteNode(curr->left, nisn);
    } else if(nisn > curr->nisn) {
        curr->right = deleteNode(curr->right, nisn);
    } else {
        deleteFound = true;
        if(curr->left == NULL && curr->right == NULL) {
            free(curr);
            return NULL;
        }
        if(curr->left == NULL) {
            struct Node* temp = curr->right;
            free(curr);
            return temp;
        } else if(curr->right == NULL) {
            struct Node* temp = curr->left;
            free(curr);
            return temp;
        }

        struct Node* is = curr->right;
        while(is->left != NULL) {
            is = is->left;
        }

```

```c
            strcpy(curr->name, is->name);
            strcpy(curr->classes, is->classes);
            strcpy(curr->jurusan, is->jurusan);
            curr->nisn = is->nisn;
            strcpy(curr->born, is->born);
            curr->grad = is->grad;

            curr->right = deleteNode(curr->right, is->nisn);
        }

        curr->height = setHeight(curr);

        int bf = getBalanceFactor(curr);

        if(bf > 1) {
            if(getBalanceFactor(curr->left) >= 0) curr = rotateRight(curr);
            else {
                curr->left = rotateLeft(curr->left);
                curr = rotateRight(curr);
            }
        } else if (bf < -1) {
            if(getBalanceFactor(curr->right) <= 0) curr = rotateLeft(curr);
            else {
                curr->right = rotateRight(curr->right);
                curr = rotateLeft(curr);
            }
        }

        return curr;
    }

void printRow(int nisn, char name[], char born[], char classes[], char jurusan[], int grad) {
    // function ini untuk print satuan data
    printf("| %d | %-25s | %-10s | %-5s | %-5s | %-15d |\n", nisn, name, born, classes, jurusan, grad);
    printf("-----------------------------------------------------------------------------------------------\n");
}

void viewMajor(struct Node* curr, char jurusan[], int *countMajor) {
    //untuk cek view jurusan dan hitung ada berapa jurusan ipa/ips
    if (curr == NULL) return;

    viewMajor(curr->left, jurusan, countMajor);

    if (strcmp(curr->jurusan, jurusan) == 0) {
        printf("-----------------------------------------------------------------------------------------------\n");
        printf("| NISN       | Name                | Birth Date | Class | Major | Graduation Year |\n");
        printf("-----------------------------------------------------------------------------------------------\n");
        printRow(curr->nisn, curr->name, curr->born, curr->classes, curr->jurusan, curr->grad);
        (*countMajor)++;
    }

    viewMajor(curr->right, jurusan, countMajor);
}

void searchNISN(struct Node* curr, int nisn, int *countNISN) {
    // untuk view nisn serta cari nisn yang di input
    if (curr == NULL) return;

    searchNISN(curr->left, nisn, countNISN);

    if (curr->nisn == nisn) {
        printf("-----------------------------------------------------------------------------------------------\n");
        printf("| NISN       | Name                | Birth Date | Class | Major | Graduation Year |\n");
        printf("-----------------------------------------------------------------------------------------------\n");
        printRow(curr->nisn, curr->name, curr->born, curr->classes, curr->jurusan, curr->grad);
        (*countNISN)++;
    }

    searchNISN(curr->right, nisn, countNISN);
}
```

```c
void viewData(struct Node* curr, int choose, bool *headerPrinted) {
    //untuk view menu
    if (root == NULL) {
        printf("No data available!\n");
        return;
    }

    if (!(*headerPrinted) && (choose == 1 || choose == 2 || choose == 3)) {
        printf("-----------------------------------------------------------------------------------------------\n");
        printf("| NISN        | Name                | Birth Date | Class | Major | Graduation Year |\n");
        printf("-----------------------------------------------------------------------------------------------\n");
        *headerPrinted = true;
    }

    if (curr == NULL) return;

    if (choose == 1) { // preorder
        printRow(curr->nisn, curr->name, curr->born, curr->classes, curr->jurusan, curr->grad);
        viewData(curr->left, choose, headerPrinted);
        viewData(curr->right, choose, headerPrinted);
    } else if (choose == 2) { // inorder
        viewData(curr->left, choose, headerPrinted);
        printRow(curr->nisn, curr->name, curr->born, curr->classes, curr->jurusan, curr->grad);
        viewData(curr->right, choose, headerPrinted);
    } else if (choose == 3) { // postorder
        viewData(curr->left, choose, headerPrinted);
        viewData(curr->right, choose, headerPrinted);
        printRow(curr->nisn, curr->name, curr->born, curr->classes, curr->jurusan, curr->grad);
    } else if (choose == 4){ //choose ipa/ips
        char jurusan[21];
        int countMajor = 0;

        do{
            printf("Enter major to search [IPA | IPS]: ");
            gets(jurusan);

            if(strcmp(jurusan, "IPA") != 0 && strcmp(jurusan, "IPS") != 0) {
                printf("Invalid major entered. Please enter 'IPA' or 'IPS'\n");
                continue;
            }

            int countMajor = 0;
            printf("Students with major %s:\n", jurusan);
            viewMajor(root, jurusan, &countMajor);

            if (countMajor == 0) {
                printf("No students found with major %s.\n", jurusan);
            }
        } while(strcmp(jurusan, "IPA") != 0 && strcmp(jurusan, "IPS") !=0);

    } else if (choose == 5){ //search nisn
        int nisn;
        int countNISN = 0;
        viewData(root, 2, headerPrinted);
        printf("Enter the 10-digit NISN: ");
        scanf("%10d", &nisn); getchar();


        searchNISN(root, nisn, &countNISN);

        if (countNISN == 0) {
            printf("No students found with NISN %d.\n", nisn);
        }
    }

}
```

```c
bool validateClasses(char classes[]) { //validasi kelas harus 10-12 + A-L
    int length = strlen(classes);

    if (length != 3) return false;

    if (classes[0] != '1') return false;
    if (classes[1] < '0' || classes[1] > '2') return false;

    char huruf = classes[length - 1];
    if (huruf < 'A' || huruf > 'L') return false;

    return true;
}

bool validateBorn(char born[]) { //validasi tanggal lahir supaya dd/mm/yyy, dd 1-30, mm 1-12, yyyy diatas 1990
    if (strlen(born) != 10) return false;

    if (born[2] != '/' || born[5] != '/') return false;

    int day = (born[0] - '0') * 10 + (born[1] - '0');
    int month = (born[3] - '0') * 10 + (born[4] - '0');
    int year = (born[6] - '0') * 1000 + (born[7] - '0') * 100 + (born[8] - '0') * 10 + (born[9] - '0');

    if (day < 1 || day > 30) return false;
    if (month < 1 || month > 12) return false;
    if (year < 1990) return false;

    return true;
}

void insertMenu() { // untuk insert data ke program
    char name[51];
    char classes[21];
    char jurusan[21];
    int nisn;
    char born[41];
    int grad;

    do {
        printf("Input Name [5 s.d 50]: ");
        fgets(name, sizeof(name), stdin);
        name[strcspn(name, "\n")] = 0;
    } while (strlen(name) < 5 || strlen(name) > 50);

    do {
        printf("Input your birth date [dd/mm/yyyy]: ");
        fgets(born, sizeof(born), stdin);
        born[strcspn(born, "\n")] = 0;
    } while (!validateBorn(born));

    do {
        printf("Enter your class [10-12 + A-L] (10B || 11L || 12F): ");
        fgets(classes, sizeof(classes), stdin); //pakai fgets karena gets ada limitnya
        classes[strcspn(classes, "\n")] = 0; //untuk buang buffer di fgets
    } while (!validateClasses(classes));

    do {
        printf("Enter your major [IPA | IPS]: ");
        fgets(jurusan, sizeof(jurusan), stdin);
        jurusan[strcspn(jurusan, "\n")] = 0;
    } while (strcmp(jurusan, "IPA") != 0 && strcmp(jurusan, "IPS") != 0);
```

```c

    do {
        printf("Enter your graduation year [2006-2024]: ");
        scanf("%d", &grad); getchar();
    } while (grad < 2006 || grad > 2024);

    char yearStr[5];                // untuk buat ID NISN 4 digit awal tahun lahir, 6 sisany random
    strncpy(yearStr, born + 6, 4);
    yearStr[4] = '\0';
    srand(time(NULL)); // supaya tidak terulang angka random nya
    int random = rand() % 1000000;
    char nisnStr[11];
    sprintf(nisnStr, "%s%06d", yearStr, random);
    nisn = atoi(nisnStr); // mengubah string ke int

    root = insertNode(root, name, classes, jurusan, born, nisn, grad); //masukin data ke insert
    printf("Insert Successfully!\n");
    system("pause");
}

void deleteMenu() { //untuk menjalankan deteleting
    bool headerPrinted = false;
    viewData(root, 2, &headerPrinted);
    if(root == NULL) return;

    int nisn;
    printf("Input NISN to Delete: ");
    scanf("%d", &nisn); getchar();

    root = deleteNode(root, nisn);
    if(deleteFound) {
        deleteFound = false;
        printf("Delete successfully\n");
    } else {
        printf("Data not found\n");
    }
    system("pause");
}

int main() { //main code display
    int option = -1;
    while (option != 4) {
        system("cls");
        printf("================================\n");
        printf("= GRADUATION INFO BINUS SCHOOL =\n");
        printf("================================\n");
        printf("1. Input\n");
        printf("2. View\n");
        printf("3. Delete\n");
        printf("4. Exit\n");
        printf(">> ");
        scanf("%d", &option); getchar();

        switch (option) {
            case 1 : {
                insertMenu();
                break;
            }
            case 2 : {
                if (root == NULL) {
                    printf("No data available!\n");
                }
                else {
                    bool headerPrinted = false;
                    int choose;
```

```c
                int choose;

                system ("cls");
                printf("================================\n");
                printf("= GRADUATION INFO BINUS SCHOOL =\n");
                printf("================================\n");
                printf("Input view graduation :\n");
                printf("1. View Preorder\n");
                printf("2. View Inorder\n");
                printf("3. View Postorder\n");
                printf("4. View Certain Data\n");
                printf("5. View Spesific (Search)\n");
                printf(">> ");
                scanf("%d", &choose); getchar();
                switch (choose){
                    case 1 : {
                        viewData(root, 1, &headerPrinted);
                        break;
                    }
                    case 2 : {
                        viewData(root, 2, &headerPrinted);
                        break;
                    }
                    case 3 : {
                        viewData(root, 3, &headerPrinted);
                        break;
                    }
                    case 4 : {
                        viewData(root, 4, &headerPrinted);
                        break;
```

```c
                    }
                    case 5 : {
                        viewData(root, 5, &headerPrinted);
                        break;
                    }
                    default : {
                        printf("Input view 1-5 :\n");
                        break;
                    }
                }
            }
            system("pause");
            break;
        }
        case 3 : {
            deleteMenu();
            break;
        }
        case 4 : {
            exit(0);
            break;
        }
        default : {
            printf("Invalid Input!\n");
        }
    }
}

return 0;
}
```